Learning + Control:

Reinforcement Learning for *Unknown* Autonomous Systems

Rahul Jain

Electrical Engineering, Computer Science* & ISE* Departments

University of Southern California

IfA@50 Workshop on ``Vistas in Controls''

September 11, 2018

(*by courtesy)

Acknowledgements

Current & Former Students

Hiteshi Sharma
(USC)

Mukul Gagrani
(USC)

Dileep Kalathil
(Texas A&M)

Former Postdocs

Abhishek Gupta
(Ohio State)

W. Haskell
(NUS)

Yi Ouyang*
(UC Berkeley)

Other Collaborators

Vivek Borkar (IITB)

A. Nayyar (USC)

Peter Glynn (Stanford)

What is Reinforcement Learning?

"Reinforcement learning is learning w *tions to actions—so as to maximize a numerical reward signal. The learner is* *but instead must discover which actions yield the most reward by trying them.*

Lai-Robbins problem

In the most interesting and challenging cases, actions may affect not only the immediate reward but also the next situation and, through that, all subs *cteristics—trial-and-error search and delayed reward—are the two most important distinguishing features of reinforcement learning."*

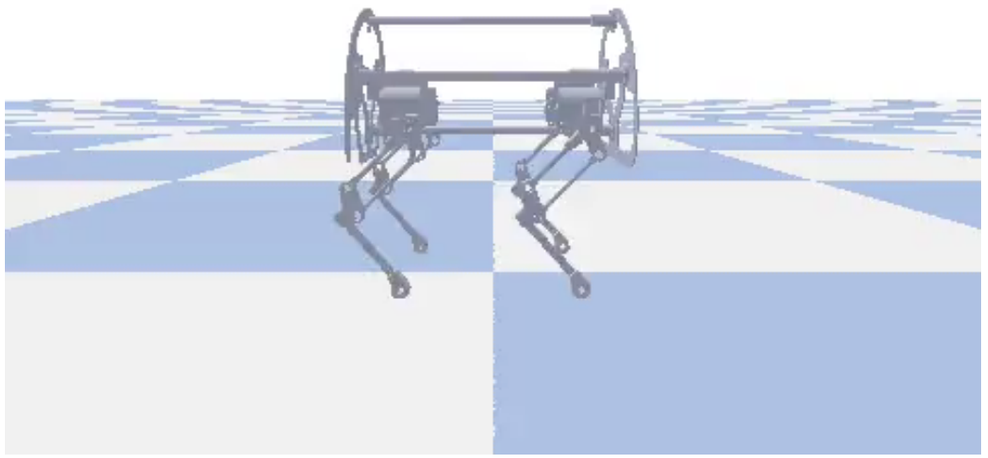
Bellman problem

-Sutton & Barto (2018)

"The exploration–exploitation dilemma has been intensively studied by mathematicians for many decades, yet remains unresolved."

-Sutton & Barto (2018)

Autonomous Robotic Control

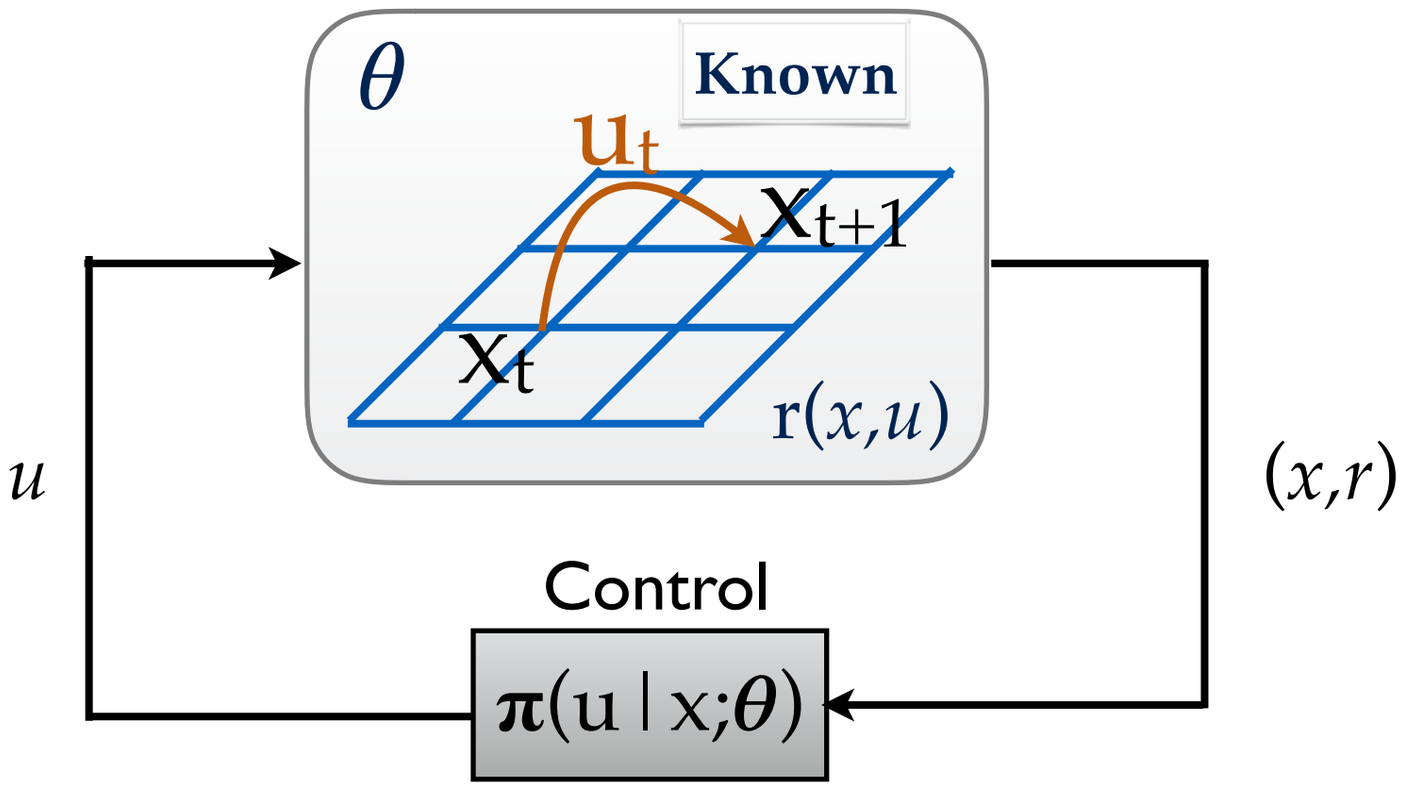


Courtesy: Bosch Center for CPS @ IISc, Bangalore

- Train algorithms in *simulation* for autonomous operation
  - The “sim2real” or “lab2real” problem
- Unknown model and Varied dynamics
  - Need to `learn to control’ *quickly* in every deployment

A general Stochastic System

$$x_{t+1} = f(x_t, u_t, w_t; \theta)$$



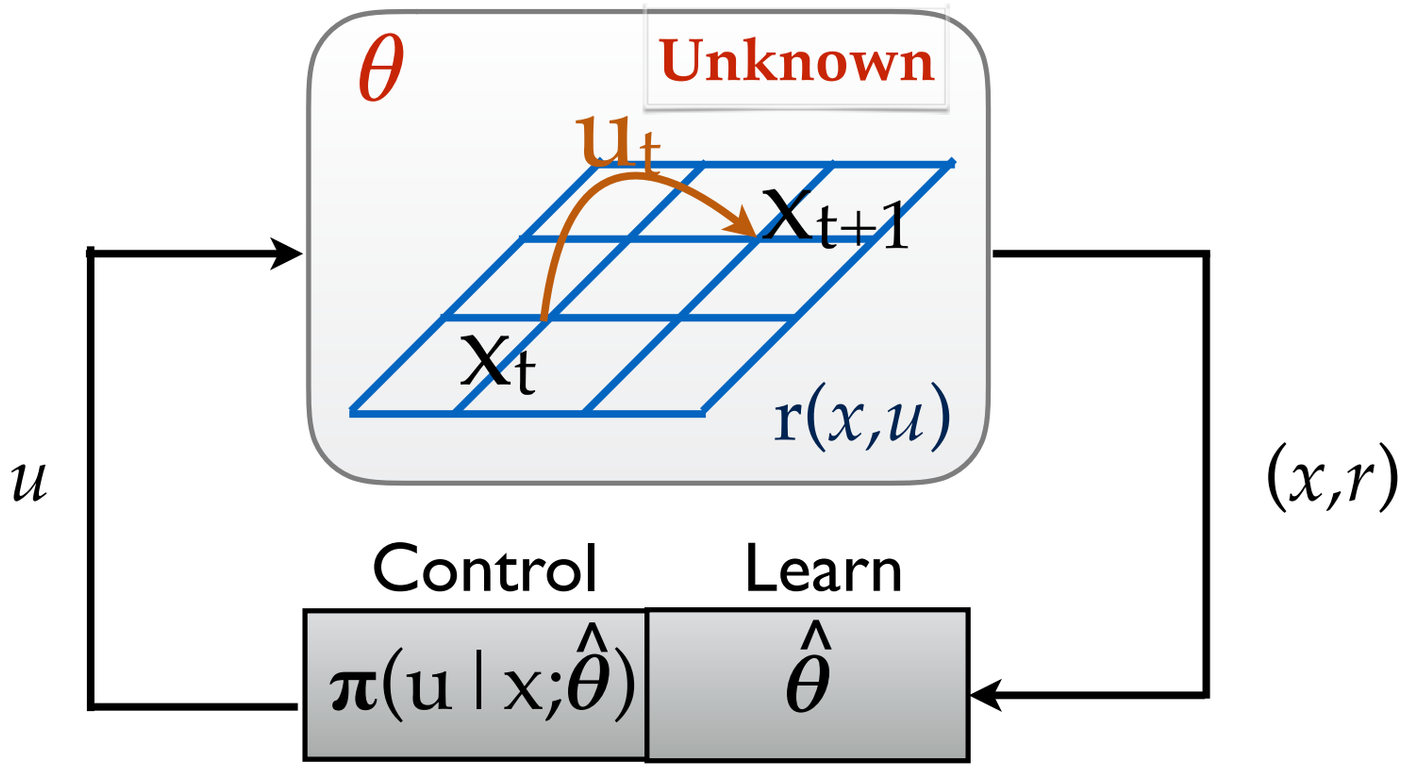
State space **X** Action space **U**

$$V_\pi(\theta) = \liminf_{T\to\infty} \frac{1}{T}\mathbb{E}[\sum_{t=1}^{T} r(x_t, u_t)]$$

Objective: Find Optimal Policy $\boldsymbol{\pi}$

Unknown Model

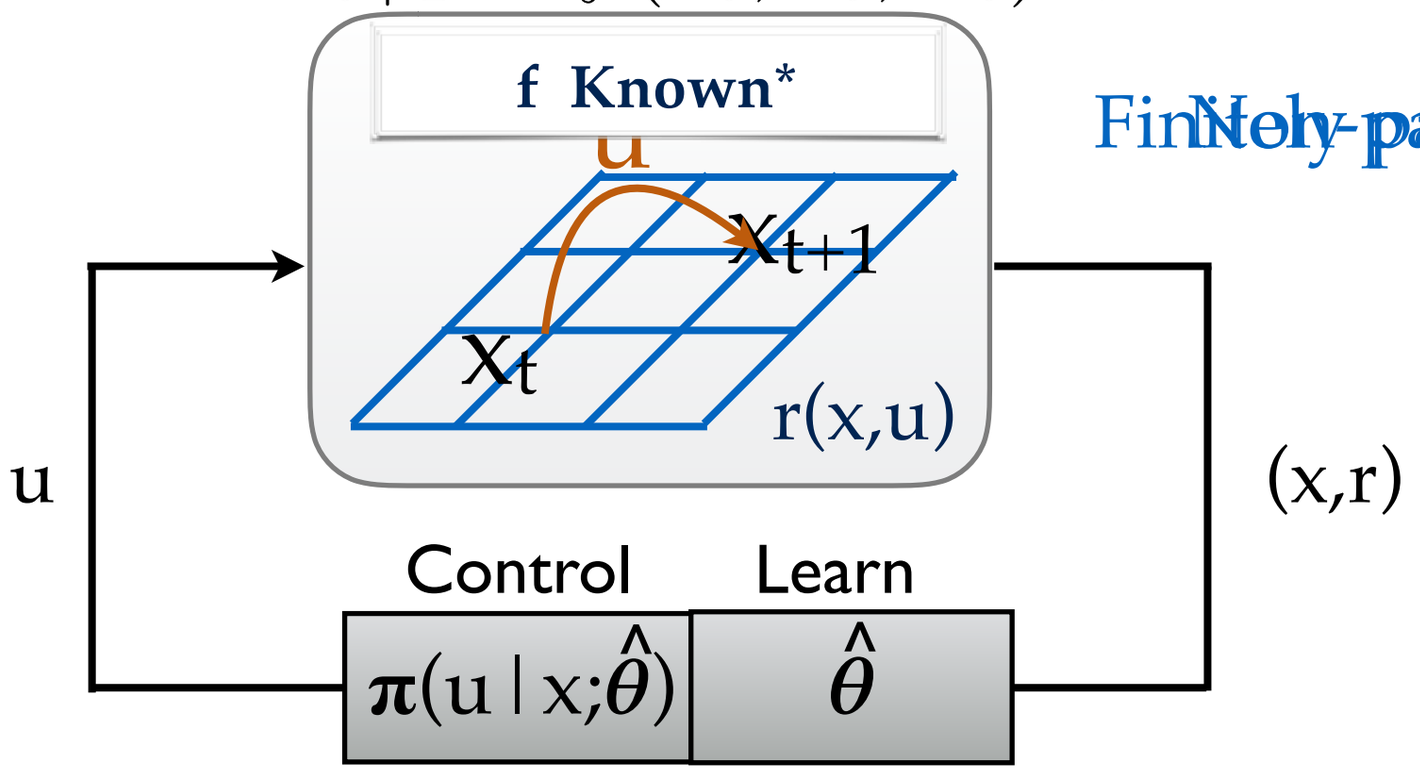
$$x_{t+1} = f(x_t, u_t, w_t; \theta)$$



- ★ True θ_o, unknown ~ prior μ
- ★ Learning policy $\phi_t(h_t)$, history h_t=(states,actions)
- ★ Objective of Learning: To find a nearly optimal policy at the fastest possible rate?

Problem 3: An (Offline) RL for Non-parametric Stochastic Systems

$$x_{t+1} \equiv f(x_t, u_t, w_t; \theta) = Ax_t + Bu_t + w_t$$



Finite-parametrized

Infinite-parametrized

MDP *Uncountable* State space **X** *Finite* Action space **U**

*only sample trajectories needed

- ★ Value of policy, $V_\pi(\theta) = \mathbb{E}[\sum_{t=1}^{\infty} \gamma^t r(x_t, u_t)] \quad \lim\inf_{T\to\infty} \frac{1}{T} \mathbb{E}[\sum_{t=1}^{T} r(x_t, u_t)] \quad x_0, u_0 \quad \gamma < 1$
- ★ Objective: $\sup_\pi \mathbf{V}_\pi$ (`Learn' Optimal Policy)

Outline

I. **Bandit Models and Online Learning**

II. An (on-policy) RL Algorithm for Unknown Systems

III. A universal (offline) RL Algorithm for *non-parametric* Stochastic Systems

Bandit Models and Online Learning

Unknown θ_1

Unknown θ_2

- Reward on Heads = \$1, on Tails =0
- *Objective*: "max expected long-term total reward"

≡

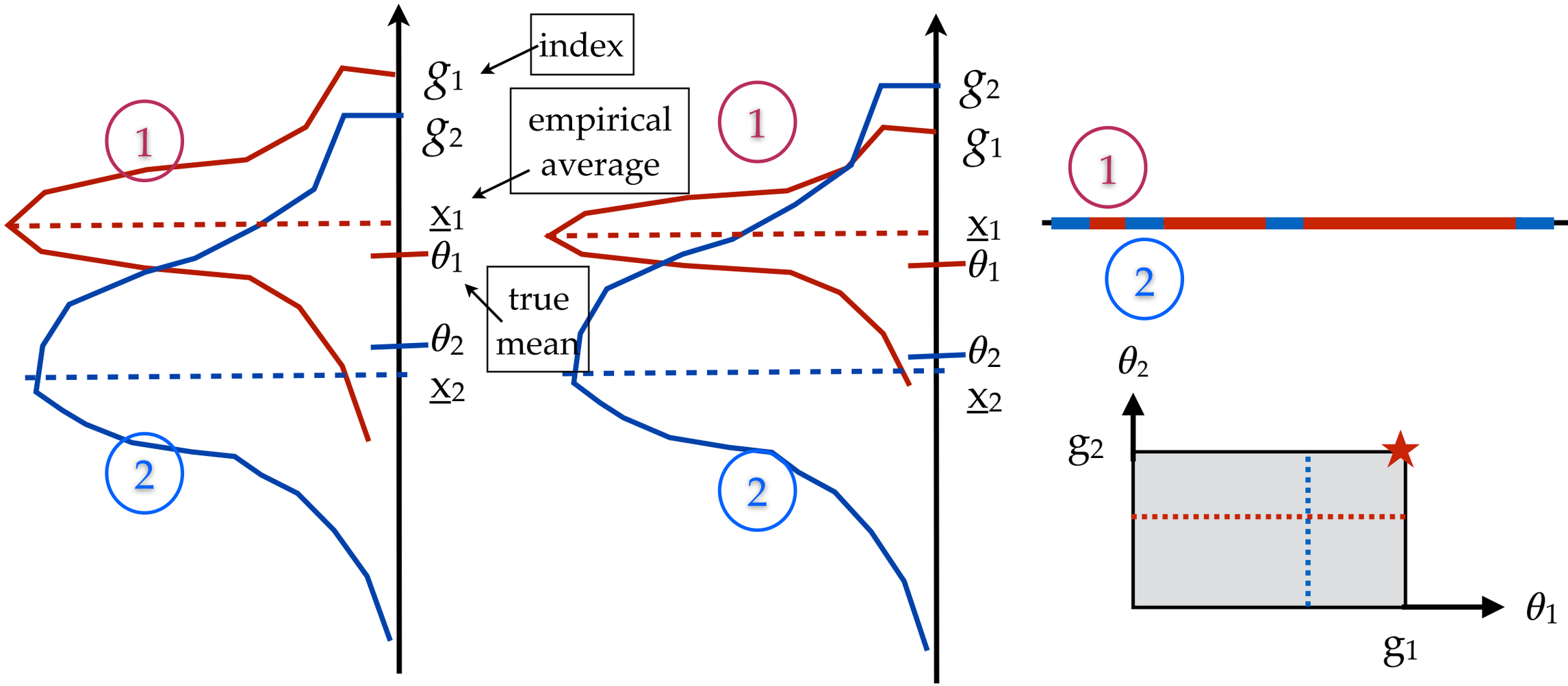
- *min* (expected) Regret

$$\mathcal{R}_T(\phi) = T\theta_{max} - \mathbb{E}[\sum_{t=1}^{T} r_t]$$

- Lai & Robbins (1985) lower bound *O(log T)*
- UCB_1 algorithm achieves *O(log T)* [Agrawal'95, Auer, et al'02]

$$g_i(t, t_i) = \underline{X}_i + \sqrt{2 \log t / t_i}$$

UCB_1 Algorithm: How does it work



Optimism in the Face of Uncertainty (OFU)

★ UCB_1 index: $$g_i(t, t_i) = \underline{X}_i(t_i) + \sqrt{\frac{2 \log t}{t_i}}$$

★ If arm i played, g_i "decreases"

- Expected Regret grows as ~ $\log T/\Delta$, optimal (by Lai & Robbins '85)

The Posterior (/Thompson) Sampling Algorithm

Unknown θ_1

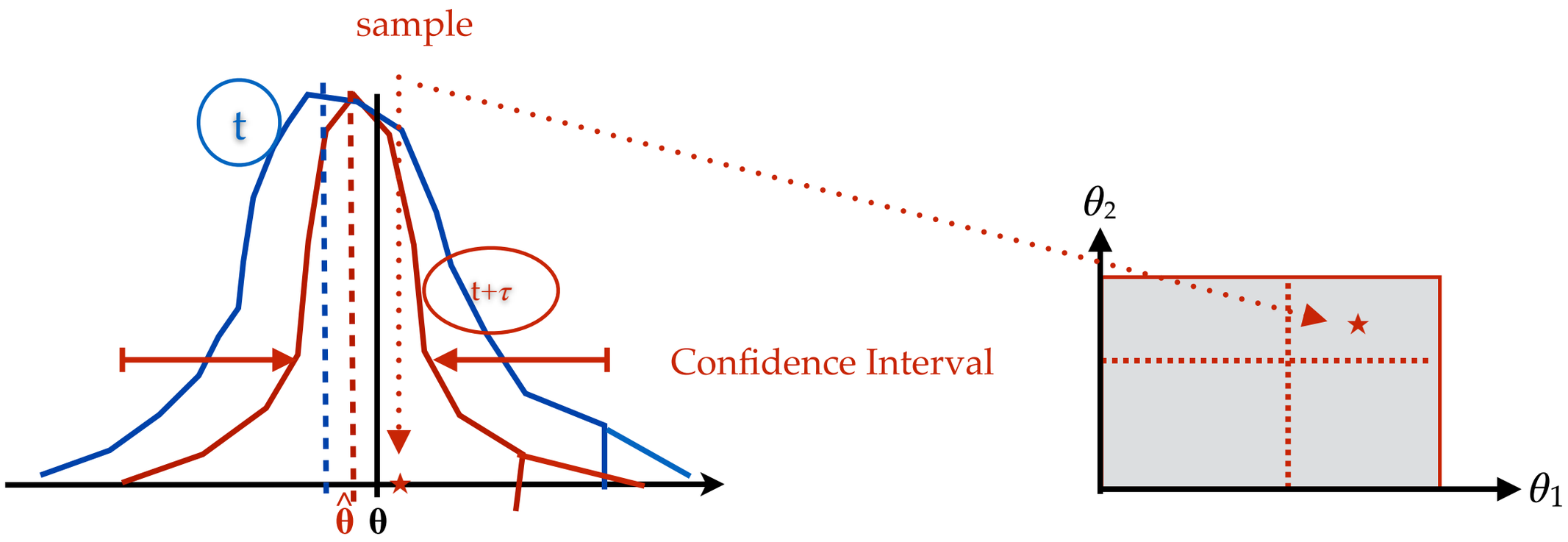
Unknown θ_2

...

Unknown θ_m

- Maintain a belief (posterior distribution), μ_i over θ_i
- Sample $\hat{\theta}_i$ from μ_i
- Choose $i^* = arg\ max_i\ \hat{\theta}_i$
- Achieves (exp) regret $R_T(\phi) = O(log\ T)$
  - *Thompson'33, Chapelle-Li'11, Agrawal-Goyal'12*
  - *Advantage: superior numerical performance, computationally simpler*

Posterior Sampling: How does it work?



Posterior P(**θ** | history)

Posterior Sampling Algorithms

Outline

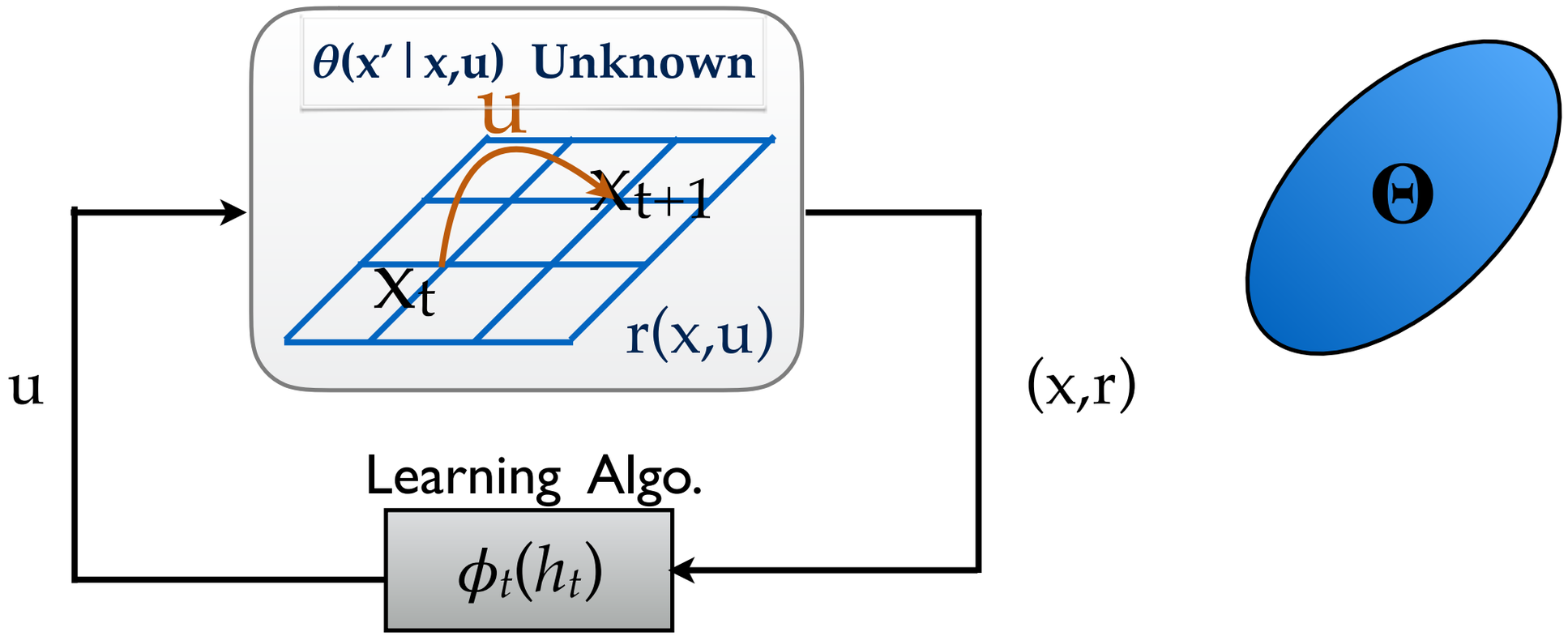
I. Bandit Models and Online Learning

II. An (on-policy) RL Algorithm for Unknown Systems

III. A universal (offline) RL Algorithm for *non-parametric* Stochastic Systems

Problem 1: Learning to Control an Unknown MDP

$$x_{t+1} = f(x_t, u_t, w_t; \theta)$$



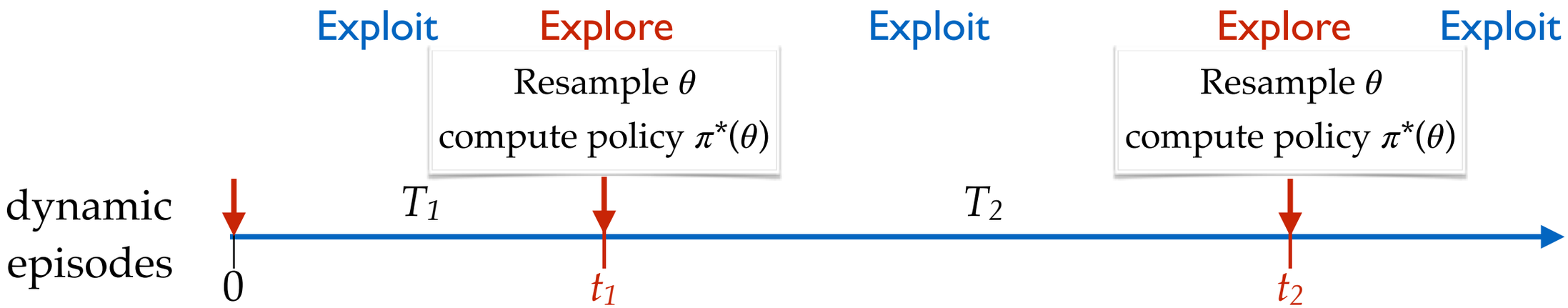
MDP *Finite* State space **X** *Finite* Action space **U**

- Learning policy $\phi_t(h_t)$ to search over space $\boldsymbol{\Theta}$
- Objective of Learning:

$$\mathcal{R}_T(\phi) = TV^*(\theta_o) - \mathbb{E}[\sum_{t=1}^{T} r(x_t, u_t)]$$

- *Lower Bound* $= \Omega(\sqrt{T})$ [Auer, et al (2009)]
- *OFU v. PS*

The **PSRL-DE** Algorithm: Posterior Sampling with Dynamic Episodes



The **PSRL-DE** Algorithm:

- ★ *Resample* θ from posterior μ_t at *start of every episode*
  - *Compute* policy optimal for sampled θ [Dithering]
- ★ *At each t,* update posterior $\mu_t(\theta) = \mathbb{P}(\theta|h_t)$ using Bayes' rule

Non-asymptotic Regret bound for PSRL-DE

Theorem. [1]

If the MDP is *weakly communicating* and its *span* ≤ H, then

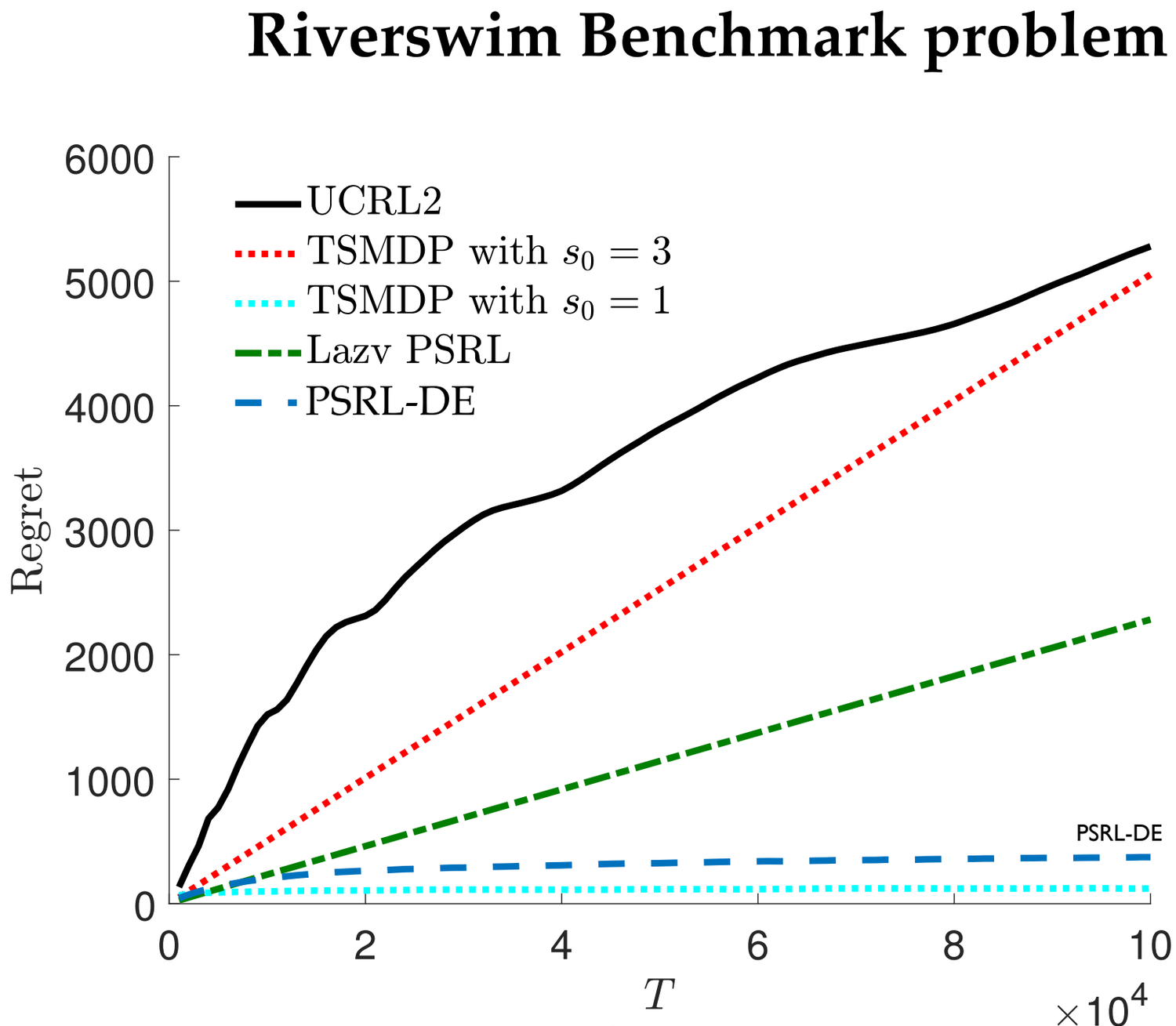
$$\mathcal{R}_T(PSRL) \leq \tilde{O}(HX\sqrt{UT})$$

where X is state space size, and U is action space size.

- ★ Up to logarithmic factors, exact constants known
- ★ PSRL-DE Algorithm works with approximately optimal policies in each episode also
- ★ Episode length can't increase faster
- ★ *Related Work:* Osband-Van Roy'14, A-Yadakori-Szepesvari'15

[1] Y. Ouyang, M. Gagrani, A. Nayyar and R. Jain, "Learning Unknown MDPs: A Thompson Sampling Approach", *NIPS, 2017*.

Numerical Performance



UCRL2: [Jaksch, Ortner, Auer (2010)]

TSMDP: [Gopalan & Mannor (2015)]

Lazy-PSRL: [Yadkori & Szepesvari (2015)]

Proof Outline

- For any function f and RV X, algorithm must satisfy

$$E[f(\theta_k, X)] = E[f(\theta_o, X)]$$

- Upper bounds number of episodes

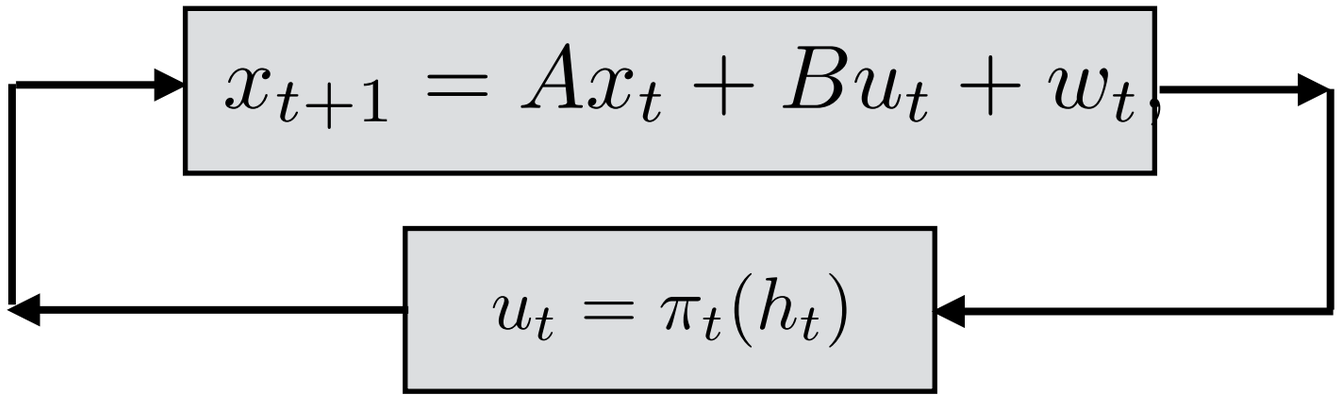
$$K_T \leq \sqrt{2XUT \log T}$$

- Upper bound between true and sampled parameters

$$T\mathbb{E}[V^*(\theta_o)] - \sum_{k=1}^{K_T} \mathbb{E}[T_k V^*(\theta_k)] \leq \mathbb{E}[K_T]$$

Problem 2: *Unknown* Stochastic Linear System

$\theta^\top = [A, B]$



$$c_t = x_t^\top Q x_t + u_t^\top R u_t$$

- ★ Parameters θ unknown
- ★ Regret $R_T(\pi) = \mathbb{E}[\sum_{t=1}^{T} c_t - TJ(\theta)]$
- ★ Optimal control policy is linear:

$$u = G(\theta)x \quad \text{where} \quad G(\theta) = -(R + B^\top S(\theta B)^{-1} B^\top S(\theta) A.$$

Assumption 1: There is a set Θ such that for all $\theta \in \Theta$, there is a unique p.d. solution to the Ricatti equation

Stochastic Adaptive Control

★ *Classical Adaptive Control…*

- Astrom-Wittenmark'94, Sastry'89, Narendra'89
- Campi and Kumar'98, Prandini-Campi'01,…

★ *Optimism in the Face of Uncertainty (OFU) (computation!)*

- Yadkori-Szepesvari'11, Ibrahimi, Javanmard, Van Roy'12,…

★ *Posterior Sampling (PS)*

- A.-Yadkori-Szepesvari'15, Abeile-Lazaric'17 ~$O(T^{2/3})$ w.h.p.

Assumption 2. For all $\theta \in \Theta$, spectral radius $\boldsymbol{\rho}(A_1+B_1G(\theta)) < \delta < 1$

Theorem. [2] Expected regret of PSRL-DE, $\mathcal{R}_T(PSRL) \leq \tilde{O}(\sqrt{T})$

[2] Y. Ouyang, M. Gagrani and R. Jain, "Learning-based Control of Unknown Linear Systems with Thompson Sampling", submitted IEEE TAC, 2017

Outline

I. Bandit Models and Online Learning

II. An (on-policy) RL Algorithm for Unknown MDPs

III. A universal (offline) RL Algorithm for *non-parametric* Stochastic Systems

Problem 3: An (Offline) RL for Non-parametric Stochastic Systems

$$x_{t+1} = f(x_t, u_t, w_t)$$

`Universal'

Computationally *simple*

Arbitrarily good approximation

Non-asymptotic (*Probabilistic*) Guarantees

★ Objective: $\boldsymbol{sup}_{\pi}\ \mathbf{V}_{\pi}$ (`Learn' Optimal Policy)

Continuous State Space MDPs

- State space Aggregation methods often don't work
- Function approximation *via* ϕ: $X \times \Theta \to \mathbb{R}$

$$V^*(x) \approx \sum_{j=1}^{J} \alpha_j \phi(x, \vartheta_j)$$

  - Approximation error depends on $d(\mathbf{\Phi}(\boldsymbol{\Theta}), V^*)$, J, basis functions picked

Randomize →

$$\inf_{\alpha,\vartheta} \frac{1}{N} \sum_{n=1}^{N} |\tilde{V}(x_n) - \sum_{j=1}^{J} \alpha_j \phi(x_n, \vartheta_j)|^2$$

Non-convex!

- (Deep) Neural Nets
  - Universal function approximators [Cybenko'89, Hornik, et al'89, Barron'93]
  - No guarantees: *How much data? How many layers/arch.? When to stop?* Lot of Computation.

Use 'Universal' Function Approx. Spaces

- ★ Use $\mathbf{\Phi}$ = RKHS (Reproducible Kernel Hilbert Space)
  - $K(x,y)$ is a cont., symm., p.d. kernel
  - $\mathscr{H}_K$ = closure of linear-span of $K(x,.)$ endowed with a natural $\langle .,. \rangle$
    - Hilbert space dense in the space of continuous functions
- ★ *Regularized L_2 Function Fitting:*

$$\inf_{f \in \mathcal{H}_K} \frac{1}{N} \sum_{n=1}^{N} |\tilde{V}(x_n) - f(x_n)|^2 + \lambda ||f||^2_{\mathcal{H}_K}$$

- ★ *By Representer Theorem, there exists a solution*

$$\hat{f}(x) = \sum_{n=1}^{N} \alpha_n K(x_n, x)$$

A 'Universal' RL Algorithm for Cont. MDPs

EVL+RKHS: A simple random basis function fitting algorithm

Key (Algorithmic) Idea:

Use *Randomization* to break Non-convexity
and
Simplify Computation

Sample Complexity of EVL+RPBF

N sampled points, J(=N) basis functions, M next states, K iterations

Theorem [3]:

Given $\epsilon \in (0, 1)$, $\delta \in (0, 1)$, select

$$N \geq N_\infty(\frac{1}{\epsilon^2}, \log\frac{1}{\delta}), \quad M \geq M_\infty(\frac{1}{\epsilon^2}), \quad K \geq K_\infty(\log\frac{1}{\delta})$$

Then,

$$||\hat{V}_k - V^*||_1 \leq \epsilon$$

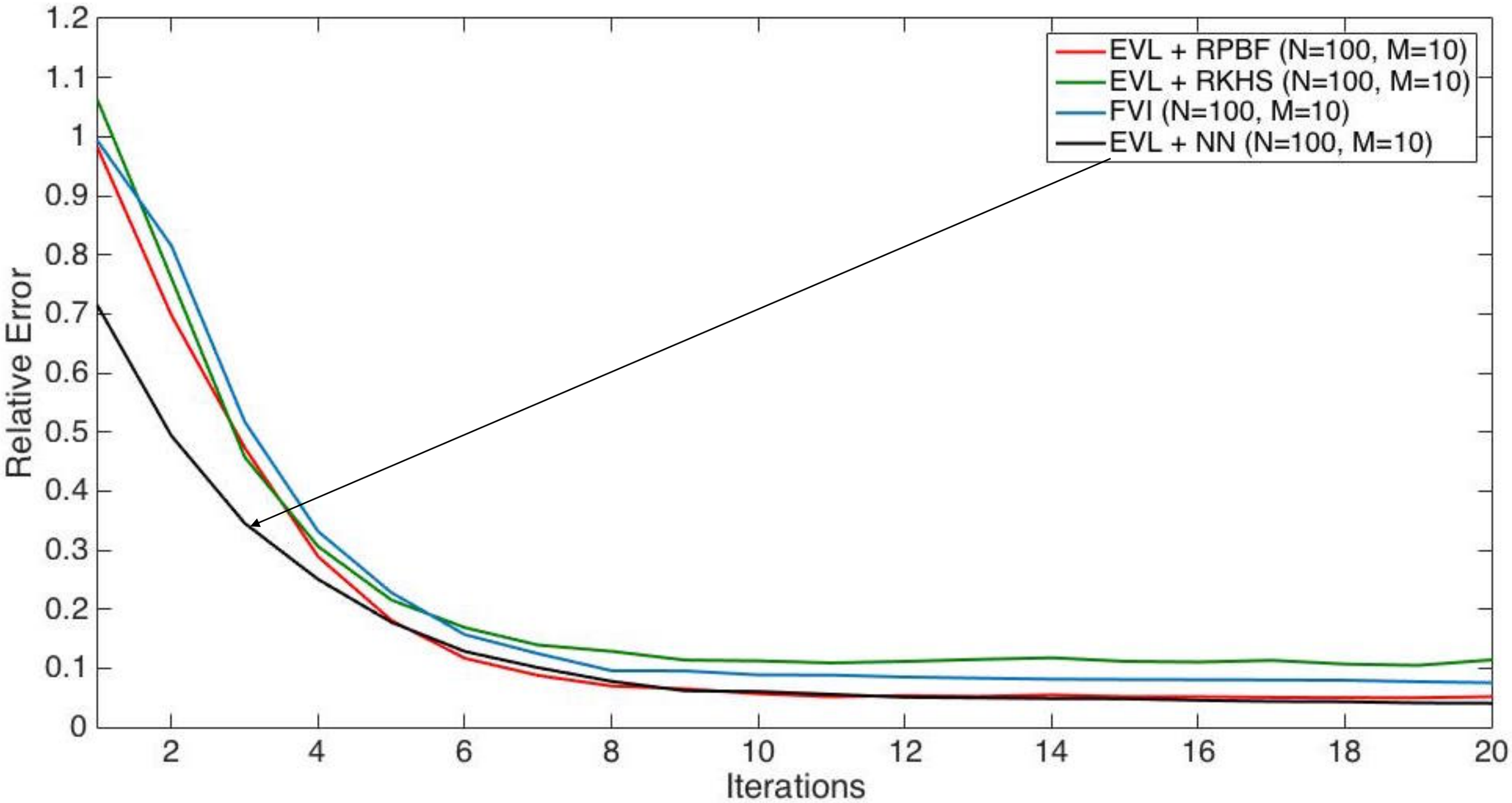
with probability > 1-δ.

- Dependence of N on ϵ is bad! but we get sup-error
- *Assumptions:* Absolute continuity of θ wrt μ and boundedness of Radon-Nikodym derivative $d\theta/d\mu$ needed!
- *Proof:* Randomized function fitting error concentration + Probabilistic Contraction Analysis of Iterated Random Operators

[3] W. Haskell, R. Jain, H. Sharma and P. Yu,,``An empirical dynamic programming algorithm for continuous state space MDPs", *submitted to IEEE TAC*

Numerical Evidence

Optimal replacement problem



RL: Challenges

- RL Literature has focused on discrete (finite) state and action spaces
  - Continuous state and action space problems are way harder
- Online R. Learning for continuous state (and action) spaces needs ideas beyond Posterior Sampling
  - Search over Value function space
- RL with constraints?
- Formal RL for safety-critical applications
- Multi-Agent RL